

Implementation of Adaptive Mesh Refinement into an Eulerian Hydrocode

David L. Littlefield

J. Tinsley Oden

Graham F. Carey

Texas Institute for Computational and Applied Mathematics

The University of Texas at Austin

Austin TX 78712

dlittlefield@iat.utexas.edu

Abstract

In this paper the implementation of an adaptive mesh refinement scheme into an existing three-dimensional Eulerian hydrocode is described. The adaptive strategy is block-based, which was required in order to leave the existing data structure intact. The focus of this work is on the development of refinement and unrefinement procedures that are conservative and preserve the locations of material interfaces, as well as error indicators suitable for use in an Eulerian hydrocode.

Introduction

Adaptive mesh refinement refers to a scheme for finite difference and finite element codes wherein the size and distribution of the computational mesh is changed dynamically so that the solution complies with some specific constraint. There are many different types of constraints of interest, depending on the goals of the computation, but a common constraint is that the error be held constant over the entire computational mesh. There are many advantages to adaptive refinement schemes; most importantly is that problems are solved in the most computational time and memory-efficient manner.

The benefits of adaptive mesh refinement have already been demonstrated in many application areas (for example, linear elasticity, gas dynamics, and acoustics), but adaptive refinement for shock wave physics codes is still a new area of research. Eulerian codes in particular may benefit significantly from adaptive mesh refinement since typically it is necessary to include a large number of elements in a simulation, even in regions where there may not be any material initially.

In this paper the implementation of an adaptive mesh refinement scheme into an existing three-dimensional Eulerian hydrocode is described. The adaptive strategy is block-based, which was required in order to leave the existing data structure intact. The focus of this work is on error indicators suitable for use in an Eulerian hydrocode, as well as refinement and unrefinement procedures.

Overview of Adaptive Strategy

When implementing adaptive refinement into an existing code, it is very important to consider the organization and data structure of the target application code. In the present case, the data is organized in (I,J,K) logical blocks that correspond to the mesh used in the problem, as is shown in Fig. 1. Within a block, the mesh contour lines must remain parallel to the coordinate axes and constrained nodes are not permitted. However, different values of I , J , and/or K are permitted in adjacent blocks. Thus a reasonable approach for implementation of adaptivity, which preserved the existing data structure in the code, was to limit refinement to the block level. Furthermore, in order to simplify the refinement process as well as communication between blocks, the refinement/unrefinement was limited to isotropic 2:1 ratios along adjacent blocks. This is illustrated in Fig. 2, where a set of communicating blocks is shown. Ghost cells are incorporated along the edges of the block, and the contents of these cells comes from a cell combine or split as is needed from the adjacent block. In this way, each block *sees* exactly the information it expects to see.

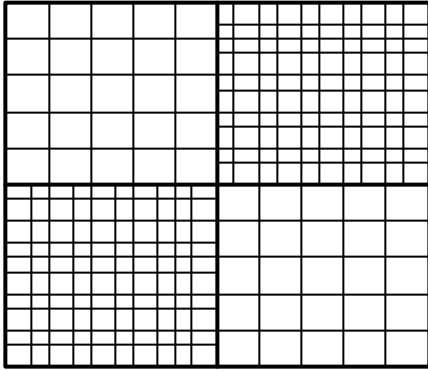


Figure 1. Organization of data in target application code

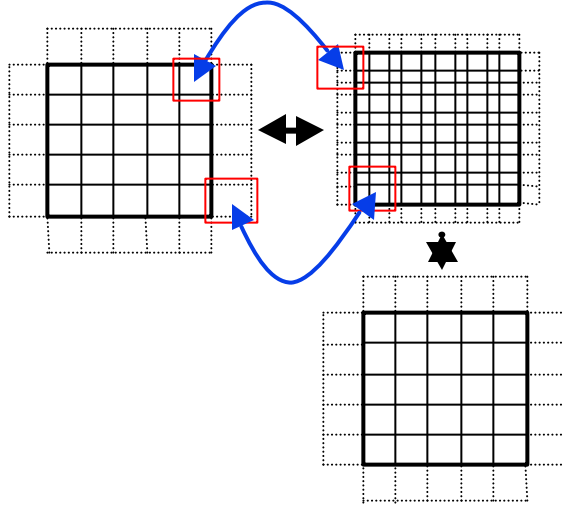


Figure 2. Block-adaptive strategy applied to target application code.

A significant part of this effort is to establish the two-way communication between blocks, as well as to make the scheme work in parallel, which is the subject of another paper [1]. The focus of this work is on the development of refinement and unrefinement schemes, as well as error indicators, suitable for implementation into a three-dimensional Eulerian shock physics code.

Refinement and Unrefinement

Efficient and accurate schemes for refinement and unrefinement of the cell variables are a crucial element in any adaptive scheme. The refinement and unrefinement strategies used here take advantage of the 1:2 and 2:1 ratios between parent and child cells, allowing these processes to be carried out rapidly and in a conservative manner.

Unrefinement Procedure

The collapse of a set of eight child cells into a single parent cell is a simple process. A method for this was implemented by Crawford [1] and was not initially modified any further. It conserves mass and energy; momentum is also conserved if the velocity profiles are linear across the parent cell. Further details of the procedure may be found in [2].

Refinement Procedure

The refinement of a parent cell into eight child cells, on the other hand, is complicated by the fact that there may be material interfaces in the parent cell. The interfaces must be reconstructed to properly map the cell variables to the children. There are already algorithms in place in the target application code that perform rezoning that could have been used for this; however, these algorithms rely in one-dimensional advection and were thought to be too dispersive given the large number of times the refinement routines are used during a typical calculation. Instead, we can take advantage of the exact geometric overlaps that exist when the refinement is limited to 2 to 1 ratios, and reconstruct the material interfaces in the child cells exactly. This eliminates dispersion errors.

Review of Youngs' Method for Interface Reconstruction

A key element in the refinement process is the proper mapping of material interfaces when elements are refined. In order to understand this process, it is useful to review Youngs' algorithm for interface reconstruction [3]. The method is a systematic approach for a unique determination of a planar interface separating two materials in a cell, given the volume fractions of the materials in the cell. Conceptually, the method is simple to understand. Where it provides the greatest benefit is by minimizing the number of possible intersection conditions that must be checked when a plane of arbitrary orientation passes through a cell (there are only five when this method is applied).

The basic strategy in the Youngs' algorithm is to determine the outward unit normal vector \mathbf{n} separating the material of interest from the other materials, and the distance d from the interface plane to a reference corner, measured along a direction parallel to \mathbf{n} . If there are only two materials in the cell and the interface plane is assumed to be planar, these two quantities uniquely define the location of the interface plane. There are five possible intersection conditions. These are given in Figure 3, and include the triangle section, quadrilateral section A, pentagonal section, hexagonal section, and the quadrilateral section B. From this comes the interface geometry as well as a value for d . A summary of the relationships needed to determine d for each of the intersection conditions can be found in [3].

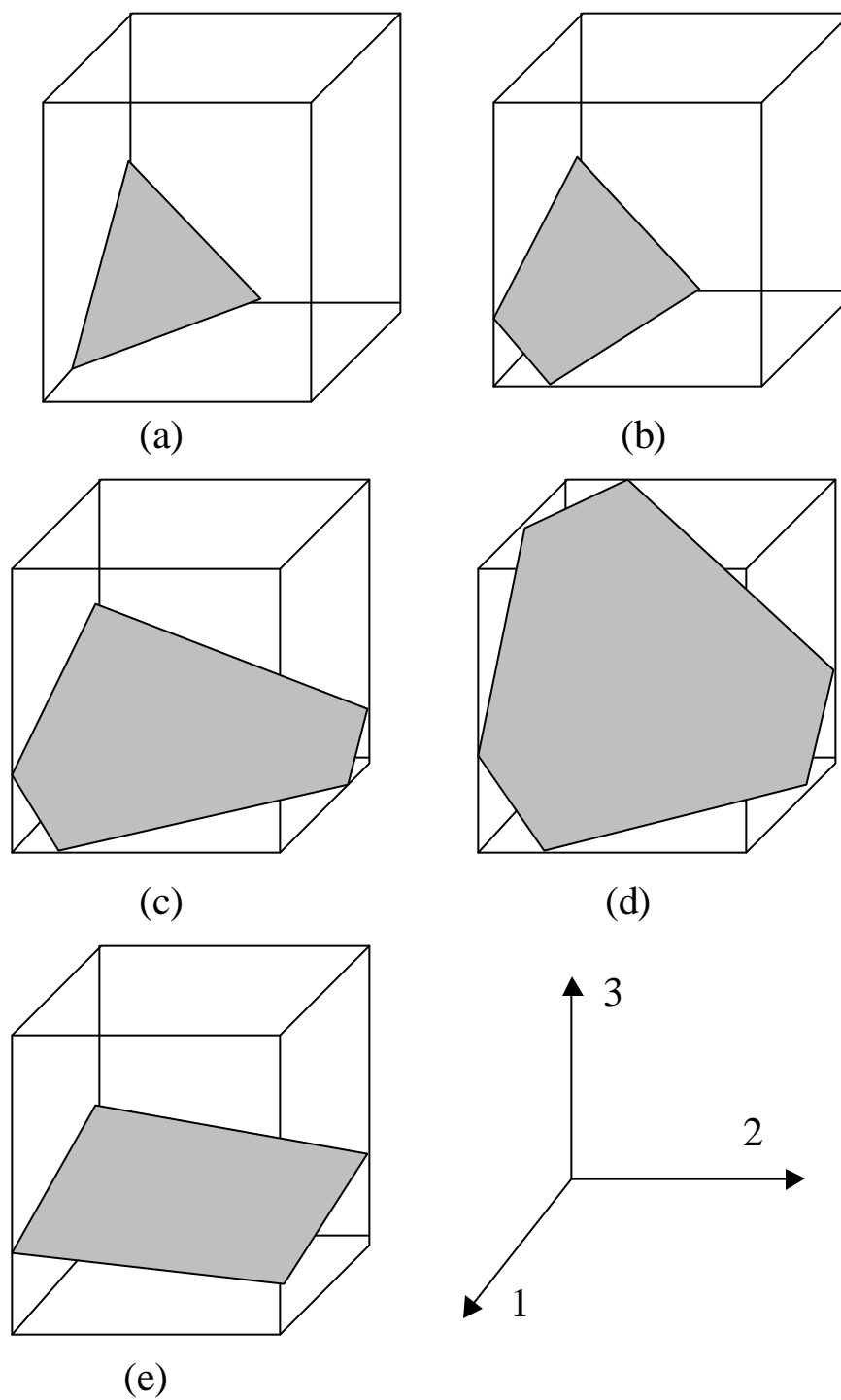


Figure 3. Possible intersection conditions for a plane intersecting a unit cube: (a) triangle section, (b) quadrilateral section A, (c) pentagonal section, (d) hexagonal section, and (e) quadrilateral section B.

Extension of Youngs' Method for Cell Refinement

When extended, Youngs' method can also be applied to cell refinement. The first step is to apply the method as originally outlined to determine d and \mathbf{n} for a particular parent cell. Once d and \mathbf{n} are known, refinement into eight equal volume child cells can be accomplished in a manner that preserves the interface reconstruction in the children. The basic procedure is to compute d and \mathbf{n} for each of the children, locate the position of the interface plane then determine the value for the volume fraction corresponding to this plane location. Consider the refinement of a parent cell into eight children, in the 1-2-3 coordinate system, as is depicted in Fig. 4.

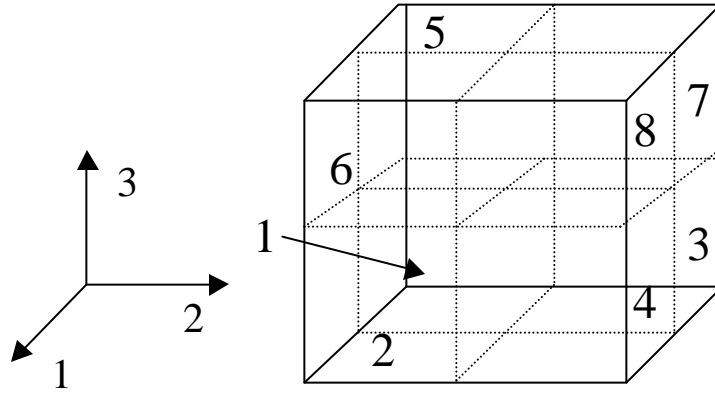


Figure 4. Schematic of refinement of a parent cell into eight equal volume child cells.

The unit normal vector to the interface plane is the same for the children as it is for the parent cell. The corner distances d_i for each of the children is easily derived and is given by

$$\begin{aligned}
 d_1 &= 2d, \\
 d_2 &= 2d - n_1, \\
 d_3 &= 2d - n_2, \\
 d_4 &= 2d - (n_1 + n_2), \\
 d_5 &= 2d - n_3, \\
 d_6 &= 2d - (n_1 + n_3), \\
 d_7 &= 2d - (n_2 + n_3), \\
 d_8 &= 2d - (n_1 + n_2 + n_3).
 \end{aligned} \tag{1}$$

Note that the values of d_i for the children are based on values for a unit cell. Now, given the value for d and \mathbf{n} in each of the child cells, the interface can be reconstructed and the required volume fraction can be computed. The number of possible intersections that must be considered is again reduced to the five possibilities given in Figure 3. A summary of the required relationships can be found elsewhere [2].

Refinement of Cell Variables

Once the volume fractions have been properly mapped from the parent cell to the children, the cell variables can then be mapped. The mapping employed assures conservation of mass, momentum and internal energy between the parent and its children. Further details of the procedure may be found in [2].

Implementation

FORTRAN routines were written to perform the refinement process described in the previous sections. Using a simple driver program, the routines were all tested. An example of the output from the testing is given in Fig. 5. Shown in Fig. 5a are volume fractions for three materials in an 8x8 two-dimensional block. In Fig. 5b, the volume fractions for the refined cells corresponding to the lower left-hand quadrant of this block are shown. Note that the volume fraction for Material 1 in refined cell (3,4) appears to be quite large. This distortion is a correct result and is caused by the fact that the interface edges do not match up along cells adjacent to parent cell (2,2) (which is used to determine the volume fractions in child cell (3,4)). Recall that the interface

Material 1

8	1.0000	0.2500	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
7	1.0000	0.2500	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
6	1.0000	0.2500	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
5	1.0000	0.2500	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
4	1.0000	0.2500	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
3	1.0000	0.2500	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
2	0.2500	0.2500	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	1	2	3	4	5	6	7	8

Material 2

8	0.0000	0.2500	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
7	0.0000	0.2500	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
6	0.0000	0.2500	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
5	0.0000	0.2500	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
4	0.0000	0.2500	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
3	0.0000	0.2500	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
2	0.2500	0.2500	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	1	2	3	4	5	6	7	8

Material 3

8	0.0000	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
7	0.0000	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
6	0.0000	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
5	0.0000	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
4	0.0000	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
3	0.0000	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
2	0.5000	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	1	2	3	4	5	6	7	8

Figure 5a. Example of volume fractions for an 8x8 parent cell.

Material 1

8	1.0000	1.0000	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000
7	1.0000	1.0000	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000
6	1.0000	1.0000	0.6154	0.0000	0.0000	0.0000	0.0000	0.0000
5	1.0000	1.0000	0.3846	0.0000	0.0000	0.0000	0.0000	0.0000
4	0.6154	0.3846	0.8284	0.0858	0.0000	0.0000	0.0000	0.0000
3	0.0000	0.0000	0.0858	0.0000	0.0000	0.0000	0.0000	0.0000
2	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	1	2	3	4	5	6	7	8

Material 2

8	0.0000	0.0000	0.0000	0.5000	1.0000	1.0000	1.0000	1.0000
7	0.0000	0.0000	0.0000	0.5000	1.0000	1.0000	1.0000	1.0000
6	0.0000	0.0000	0.0000	0.4667	1.0000	1.0000	1.0000	1.0000
5	0.0000	0.0000	0.0000	0.5333	1.0000	1.0000	1.0000	1.0000
4	0.0000	0.0000	0.0000	0.0858	1.0000	1.0000	1.0000	1.0000
3	0.4667	0.5333	0.0858	0.8284	1.0000	1.0000	1.0000	1.0000
2	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	1	2	3	4	5	6	7	8

Material 3

8	0.0000	0.0000	0.5000	0.5000	0.0000	0.0000	0.0000	0.0000
7	0.0000	0.0000	0.5000	0.5000	0.0000	0.0000	0.0000	0.0000
6	0.0000	0.0000	0.3846	0.5333	0.0000	0.0000	0.0000	0.0000
5	0.0000	0.0000	0.6154	0.4667	0.0000	0.0000	0.0000	0.0000
4	0.3846	0.6154	0.1716	0.8284	0.0000	0.0000	0.0000	0.0000
3	0.5333	0.4667	0.8284	0.1716	0.0000	0.0000	0.0000	0.0000
2	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	1	2	3	4	5	6	7	8

Figure 5b. Computed volume fractions for the 8x8 child cells corresponding to the lower left-hand quadrant.

reconstruction method described requires that the interfaces be planar and does not require that the planes match up along the edges of adjacent cells.

Error Estimation

A key component of any adaptive scheme is the ability to estimate the error in the calculation accurately and efficiently. Decisions are made regarding refinement and/or unrefinement based on values computed for the error estimates. There are numerous techniques that have been used successfully as indicators for the error in numerical solutions of differential equations. *A posteriori* error estimation methods for differential equations have been under continual development for about the last two decades. A summary of the current state-of-the-art in posteriori error estimation can be found in [4]. However, very few methods have been examined and tested for highly nonlinear equations, which exist in shock physics and large deformation applications.

The complexity of the governing equations, when coupled with the speed of time integration in the explicit calculation used for these applications, prohibits detailed estimates of the error in the solution. However, residuals associated with each of the governing equations can be estimated very quickly, and do not represent a significant computational burden over and above the time integration itself. Then, hopefully the residuals will provide meaningful information for use in driving an adaptive process. This is the approach considered here. Furthermore, although the numerical integration procedure used in the target application code is based on finite volume, the formulation developed here for the residual estimates is based on finite elements, using the solution from code as a finite element approximation to the exact solution. In this way we can build on the residual estimation procedures that have already been developed for the finite element method.

Element-by element residuals from each of the governing conservation equations solved in the application code can be formulated as

$$\sum_k \left\{ \int_{\Omega_k} w \left[\frac{\partial \mathbf{r}_x}{\partial t} - \nabla \cdot \mathbf{r}_x \mathbf{v}_x \right] d\Omega - \frac{1}{2} \int_{\partial\Omega_k \setminus \Gamma} w \mathbf{v}_{nx} [\mathbf{r}_x] d\Gamma \right\} = \sum_k \int_{\Omega_k} w R_r d\Omega, \quad (2)$$

$$\begin{aligned} & \sum_k \left[\int_{\Omega_k} \left\{ \mathbf{w} \cdot \mathbf{r}_x \left[\frac{\partial \mathbf{v}_x}{\partial t} + \mathbf{v}_x \cdot \nabla \mathbf{v}_x \right] + \mathbf{w} \cdot \nabla \cdot (-p_x \mathbf{I} + \mathbf{t}_x) \right\} d\Omega + \frac{1}{2} \int_{\partial\Omega_k \setminus \Gamma} \mathbf{w} \cdot [-p_x \mathbf{I} + \mathbf{t}_x] \cdot \mathbf{n} d\Gamma \right] \\ & = \sum_k \int_{\Omega_k} \mathbf{w} \cdot \mathbf{R}_v d\Omega, \end{aligned} \quad (3)$$

and

$$\begin{aligned} & \sum_k \left[\int_{\Omega_k} \left\{ w \mathbf{r}_x \left[\frac{\partial e_x}{\partial t} + \mathbf{v}_x \cdot \nabla e_x \right] - w \nabla \cdot [(-p_x \mathbf{I} + \mathbf{t}_x) \cdot \mathbf{v}_x] \right\} d\Omega + \frac{1}{2} \int_{\partial\Omega_k \setminus \Gamma} w \{ [-p_x \mathbf{I} + \mathbf{t}_x] \cdot \mathbf{v}_x \} \cdot \mathbf{n} d\Gamma \right] \\ & = \sum_k \int_{\Omega_k} w R_e d\Omega, \end{aligned} \quad (4)$$

where \mathbf{r} is the density, \mathbf{v} the velocity, p the pressure, \mathbf{t} the deviatoric stress tensor, e the internal energy, and the x subscript denotes the approximate solution. The function w and its vector counterpart \mathbf{w} are arbitrary weight functions, Ω the control volume and Γ its surface, and R_r , \mathbf{R}_v , and R_e are the residuals associated with mass, momentum and energy conservation, respectively. Here, the sums over k denote sums over all the elements, and the terms in $[\cdot]$ for the surface integrals denote the jump in the specified quantity across the element edge. The weight function in each of these integrands is arbitrary, but for error estimation purposes it is usually sufficient to estimate the magnitudes of the integrals with respect to some norm. In the present case, we use the L_2 norm, which allows us to write the residuals as

$$C_r \sum_k |w|_{H_1(\Omega_k)} \left\{ h_k \|r_{ir}\|_{L_2(\Omega_k)} + \frac{1}{2} h_k^{1/2} \|r_{br}\|_{L_2(\partial\Omega_k)} \right\} \geq \sum_k \int_{\Omega_k} w R_r d\Omega, \quad (5)$$

$$C_v \sum_k |w|_{H_1(\Omega_k)} \left\{ h_k \|r_{iv}\|_{L_2(\Omega_k)} + \frac{1}{2} h_k^{1/2} \|r_{bv}\|_{L_2(\partial\Omega_k)} \right\} \geq \sum_k \int_{\Omega_k} \mathbf{w} \cdot \mathbf{R}_v d\Omega, \quad (6)$$

and

$$C_e \sum_k |w|_{H_1(\Omega_k)} \left\{ h_k \|r_{ie}\|_{L_2(\Omega_k)} + \frac{1}{2} h_k^{1/2} \|r_{be}\|_{L_2(\partial\Omega_k)} \right\} \geq \sum_k \int_{\Omega_k} w R_e d\Omega, \quad (7)$$

where r_{ix} and r_{bx} denote the integrands from the volume and surface integrals from Eqs. (2) – (4), and are referred to as *interior* and *boundary* residuals, respectively; Ω_k is a subdomain of Ω_k consisting of elements sharing a common edge with Ω_k ; h_k is the diameter of element k ; C_i is a constant independent of the element size, and $|\cdot|$ denotes the H_1 semi-norm. Each of the terms in $\{\cdot\}$ from Eqs. (5) – (7) are readily computed from the numerical solution to the problem.

One-Dimensional Simulations

In order to test the validity of the residuals for use as error indicators, as well as accuracy as the error estimates, some one-dimensional calculations of shock propagation were performed. Shown in Fig. 6 is a description of the problem. An aluminum bar strikes a stationary aluminum bar at 2000 m/s. This causes shock waves to be propagated forward into the stationary bar and backward into the moving bar. Figure 7 shows results from a one-dimensional lagrangian simulation of the impact encounter. Shown are calculated velocity profiles at various times after impact, along with the exact solution. Agreement between the solutions is good; the numerical solution exhibits some

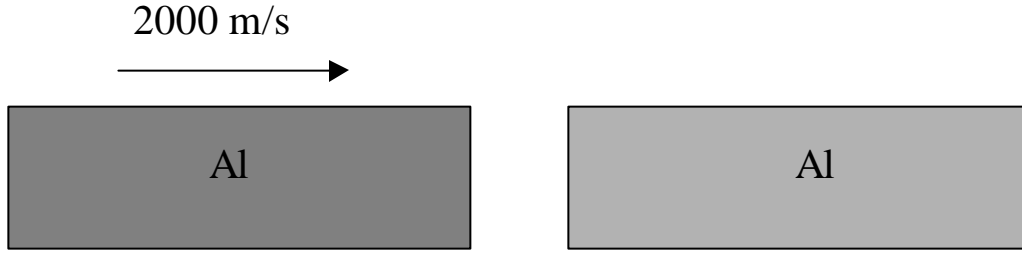


Figure 6. Setup for one-dimensional shock propagation test problem.

ringing near the shock boundaries, which is due to the artificial viscosity necessary for a valid solution. Artificial viscosity is not used in the analytical solution.

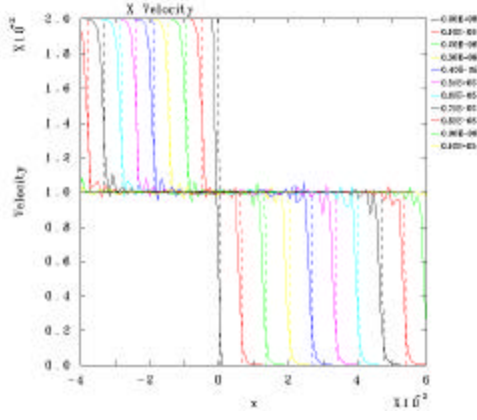


Figure 7. Velocity versus position at several times.

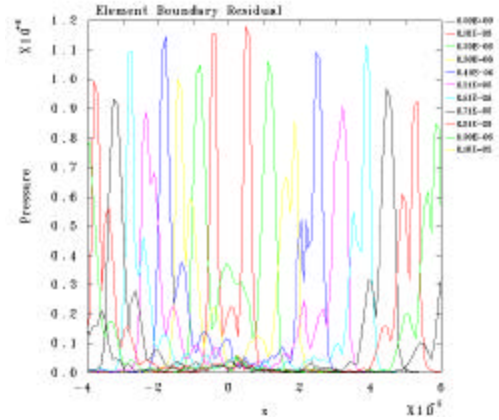


Figure 8. $1/2 h_k^{1/2} ||n_b||$ versus position at several times.

Norms of the residuals calculated from the solutions shown in Fig. 7 are shown in Figs. 8 – 10. In Fig. 8, the norm of the boundary residual from the momentum equation, multiplied by $1/2 h_k^{1/2}$, is shown. The boundary residual is largest at the shock interface. Figure 9 shows the interior residual from the momentum equation multiplied by h_k . The interior residual is also large near the shock boundary, but is smaller in magnitude than the boundary residual. The sum of these two residuals, which is the term in $\{ \}$ in Eq. (6), is given in Fig. 10.

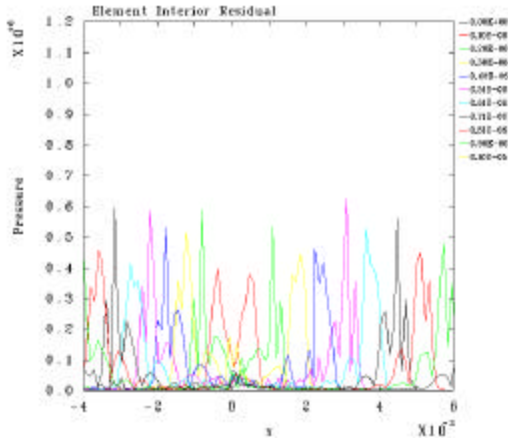


Figure 9. $h_k ||r_{in}||$ versus position at several times.

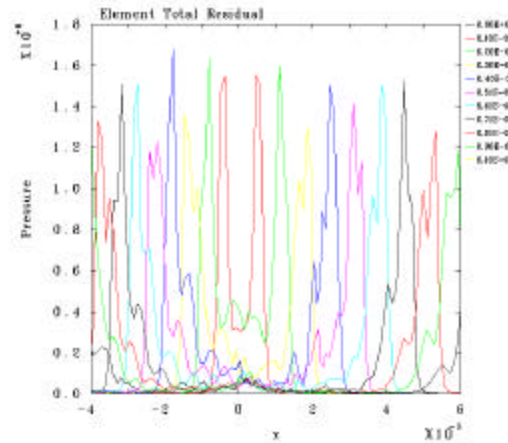


Figure 10. Residual sum versus position at several times.

Two-Dimensional Simulations

In the two-dimensional case it is usually not possible to derive an analytical solution for problems of practical significance. Nevertheless, it is possible to make qualitative judgments about the performance error indicators in picking up regions in the problem where the error in the solution is probably large (for example, across shock waves or in regions of large deformations).

A two-dimensional lagrangian simulation of a generic Taylor anvil impact was performed to test the utility of the residuals for use as an indication of the error. Shown in Fig. 11a is the problem setup. The steel cylinder, 20 mm in diameter and 40 mm long, strikes a rigid boundary at $x = 0$ with an initial velocity of 1000 m/s. The simulation was run assuming axial symmetry. Interaction with the rigid boundary produces a mushroom-shaped cylinder at subsequent times after impact, which is illustrated in Figs. 11b – 11d.

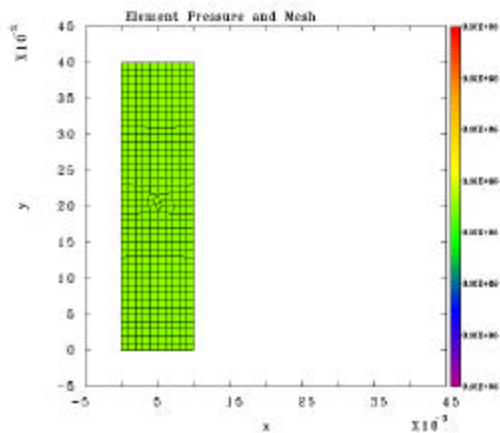


Figure 11a. Problem setup.

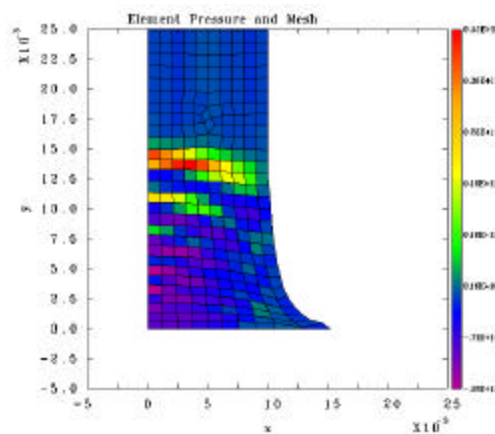


Figure 11b. Pressure contours at 3.0 ms.

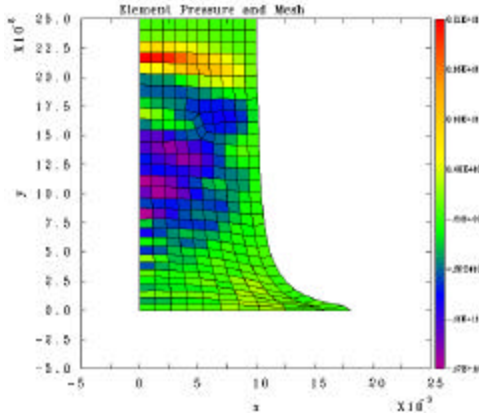


Figure 11c. Pressure at 4.9 ms.

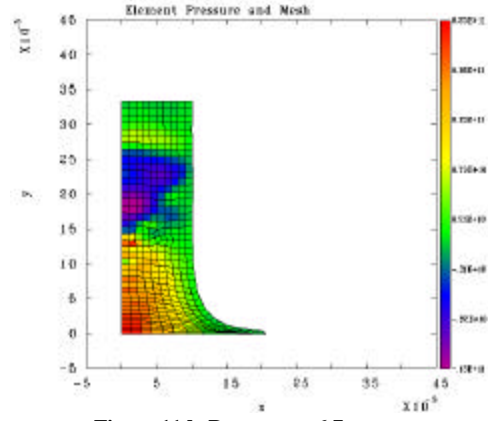


Figure 11d. Pressure at 6.7 ms.

Values for the total residual associated with the momentum equation [i.e. the term in $\{ \}$ in Eq. (6)] is given in Figs. 12a-c. Clearly, the residuals are large across the shock wave propagating back towards the back of the cylinder. The residual also starts to increase in magnitude near the tip of the mushrooming head, where the largest deformations are occurring. This is probably a region of large error, which is also indicated by the residual.

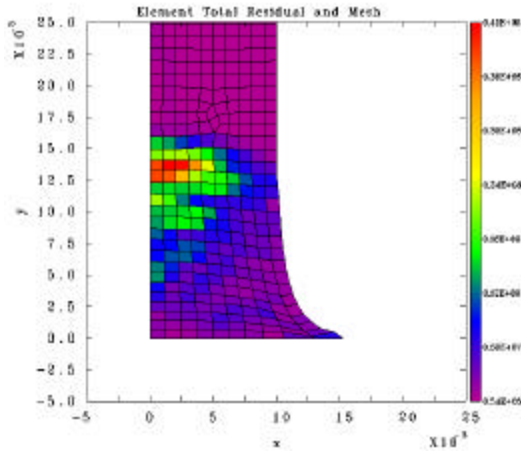


Figure 12a. Residual sum at 3.0 ms.

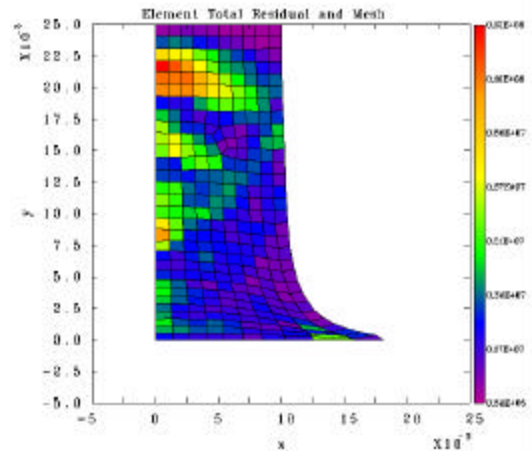


Figure 12b. Residual sum at 4.9 ms.

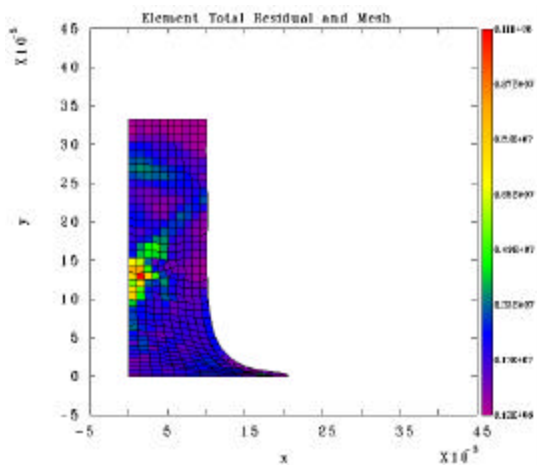


Figure 12c. Residual sum at 6.7 ns.

Acknowledgements

This work was supported by the DoD High Performance Computing Modernization Program CEWES Major Shared Resource Center through Programming Environment and Training (PET). The authors would also like to acknowledge Dr. Rick Weed for his suggestions and comments.

Conclusions

Algorithms necessary for implementation of adaptive mesh refinement into a three-dimensional Eulerian hydrocode have been described in this work. An existing technique for interface reconstruction was extended to yield the proper volume fractions for mixed-material cells during refinement. Preliminary results for error indicators are promising, but further enhancements and testing are still needed.

References

- [1] Crawford, D., "Adaptive mesh refinement in CTH", U. S. Army Symposium on Solid Mechanics, Myrtle Beach SC, April 11-14 (1999).
- [2] Littlefield, D. L. and Oden, J. T., "Implementation of Adaptive Mesh Refinement in an Eulerian Hydrocode", CEWES MSRC PET Report, in preparation.
- [3] Youngs, D., "An Interface Tracking Method for a 3D Eulerian Hydrodynamics Code", Atomic Weapons Research Establishment Report No. AWRE/44/92/35, April (1987).
- [4] Ainsworth, M. And Oden, J. T., "A Posteriori Error Estimation in Finite Element Analysis", Texas Institute for Computational and Applied Mathematics Report No. 96-19, May (1996).
- [5] Johnson, G. R. and Stryk, R. A., "User Instructions for the EPIC-3 Code", Alliant Techsystems, Inc. (1994).
- [6] Patra, A. K. And Littlefield, D. L., "A Report on the Development of Some Adaptive Methods for Computational Impact Mechanics", Institute for Advanced Technology Report, in preparation.
- [7] Zienkiewicz, O. C. and Zhu, J. C., "The Superconvergent Patch Recovery and A Posteriori Error Estimates, Part I: The Recovery Technique", *Int. J. Numer. Methods Engng.*, **33**, pp. 1331-1364, (1992).

Figure Captions

Figure 1. Organization of data in target application code.

Figure 2. Block-adaptive strategy applied to target application code.

Figure 3. Possible intersection conditions for a plane intersecting a unit cube: (a) triangle section, (b) quadrilateral section A, (c) pentagonal section, (d) hexagonal section, and (e) quadrilateral section B.

Figure 4. Schematic of refinement of a parent cell into eight equal volume child cells.

Figure 5a. Example of volume fractions for an 8x8 parent cell.

Figure 5b. Computed volume fractions for the 8x8 child cells corresponding to the lower left-hand quadrant.

Figure 6. Setup for one-dimensional shock propagation test problem.

Figure 7. Velocity versus position at several times.

Figure 8. $1/2h_k^{1/2}\|r_{bv}\|$ versus position at several times.

Figure 9. $h_k\|r_{iv}\|$ versus position at several times.

Figure 10. Residual sum versus position at several times.

Figure 11a. Problem setup.

Figure 11b. Pressure contours at 3.0 μs .

Figure 11c. Pressure at 4.9 μs .

Figure 11d. Pressure at 6.7 μs .

Figure 12a. Residual sum at 3.0 μs .

Figure 12b. Residual sum at 4.9 μs .

Figure 12c. Residual sum at 6.7 μs .